We have removed material that we did not cover. We will include material that last semester did not cover, like the material in cards.py and the testing material.

Closed book, closed note. You may not use any resources that I do not provide. Every question is multiple-choice or short answer.

```
import re
sum = 0

pattern = 'back'
if re.match(pattern, 'backup.txt'):
    sum += 4
    print('a')
if re.match(pattern, 'text.back'):
    sum += 3
    print('b')
if re.search(pattern, 'backup.txt'):
    sum += 2
    print('c')
if re.search(pattern, 'text.back'):
    sum += 1
```

```
counter = 1
def doLotsOfStuff():
        for i in (1, 2, 3):
                counter += 1
doLotsOfStuff()
print(counter)
```

1. What is the output of the program to the left?

```
for i in  range(2):
   print (i)

for i in range(4,6):
   print (i)
```

2. What is the output of the program to the left?

6. What is the output of the program above?

```
kvps  = {"user":"Chris", "password":"Becca"}
print (kvps['password'])
```

4. What is the output of this program?

```
kvps  = {"user":"Chris", "password":"Becca"}
print (kvps[0])
```

5. What is the output of this program?

```
foo = {}
print (type(foo))
```

7. What is the output of the program?

```
foo = (3, 4, 5)
print (type(foo))
```

8. What is the output of the program?

```
foo = {1:'1', 2:'2', 3:'3'}
print (len(foo))
```

9. What is the output of the program?

```
foo = {1:'1', 2:'2', 3:'3'}
del foo[1]
foo[1] = '10'
del foo[2]
print (len(foo))
```

10. What is the output of the program?

```
names = ['A', 'B', 'C', 'D']
print (names[-1][-1])
```

11. What is the output of the program?

```
numbers = [1, 2, 3, 4]
numbers.append([5,6,7,8])
print (len(numbers))
```

12. What is the output of the program?

- Given a bunch of code from cards.py and invocations of methods / print statements, what prints?
- Given regular expressions (check out the regular expressions in the slides), can you describe them in English? Match them to English descriptors?

Other questions to consider

- What do YOU think the most important part of good code documentation is? Why?
- Why define functions in Python?
- Why define classes in Python?
- What's an example of a class definition? Its methods? Its instance variables? How would you explain it in English?
- True or False: Any method can be invoked on any class instance
- True or False: Any test can be expressed with the assertEqual method

Could you write/understand code to:

Find all of the hyperlink references on a given page if you knew the URL (e.g. http://cnn.com),and:

1 - Print a listing of all of the unique values along with how many times they appear on the page.

2 Print the listing in order of appearance from least to most.

3 If the hyperlink reference is empty, store it as the string 'Empty'.

Or

Given a url and clear documentation about query parameters (filled in chart, a lot like in the iTunes search API docs, for example), make a request to an API

Or

Given a test AND a bunch of code, could you tell if the code will pass the test or not? (Use examples from your HW and lecturetests.py to study, too! How could you change code if you wanted the test to pass? What happened in your code in HW that made tests NOT pass? What was happening?)

YOU WOULD NOT HAVE TO WRITE ALL OF THIS CODE ON THE EXAM! But you might be given code that does stuff like the above and be asked questions about it – short answer questions or multiple choice questions.